

UNITED STATES PATENT APPLICATION

for

METHOD AND SYSTEM FOR FAULT DIAGNOSIS IN A DATA NETWORK

Inventors:

Denise Gürer

Elaine Lusher

Prepared by:

WAGNER, MURABITO & HAO LLP

TWO NORTH MARKET STREET

THIRD FLOOR

SAN JOSE, CALIFORNIA 95113

(408) 938-9060

# METHOD AND SYSTEM FOR FAULT DIAGNOSIS IN A DATA NETWORK

## BACKGROUND OF THE INVENTION

### FIELD OF THE INVENTION

5       The present invention relates to the field of network management. More specifically, the present invention relates to the self-diagnosis of faults within data networks.

### RELATED ART

10       Today's high speed data networks are heterogeneous, more complex and increasingly data intensive. Thus, the networks are becoming more difficult to manage due to network complexity and size. Network engineers (NEs) manage the data networks and must be familiar with their specific data  
15 network's topology, the behavior of all the devices on their network, and be able to process huge amounts of seemingly unrelated data.

20       An important activity in network management is network diagnosis. A single fault in the data network can produce hundreds and even thousands of alarms that must be analyzed by the NE, which is a prohibitive task. Traditional network fault diagnosis required the direct involvement with a NE who analyzed large amounts of seemingly unrelated fault data to  
25 determine what is causing the data network to operate improperly.

The NE necessarily must have expertise in troubleshooting, an understanding of network device behavior, and specific knowledge regarding their network, such as, topology, typical traffic conditions, applications, and legacy systems. One problem with the management of data networks is that fewer NEs with the necessary specialized expertise are available. Thus, NEs are responsible for more area within the field of network management to overcome the lack of NEs in the field. However, allocation of the resources provided by the NE is inefficient. A NE spends an inordinate amount of time monitoring and troubleshooting a data network in order to resolve problems on the network. That time could be better spent accomplishing other network management tasks.

Prior Art Figure 1 is an illustration of the traditional fault management as a process. A NE 150 has access to network data 130 and alarms 120 from a data network (e.g., local area network (LAN) 110). A network management tool can be used to monitor and collect performance data in the form of remote monitoring (RMON) data (e.g., RMON-1 and RMON-2), alarms, or events where present thresholds have been crossed. This data is aggregated and displayed to the NE in the form of display data 140, such as, graphs and tables.

Typically a troubleshooting episode is triggered by a user (not shown) of the data network 110. The user contacts

the NE 150 with a problem regarding the network 110. For example, the user may be experiencing slowness in responses transmitted through the network 110, or the user may have lost connectivity.

5

At this point, it becomes the duty of the NE to isolate the fault and take corrective action. The NE can analyze the display data 140 to manually troubleshoot the problem. Many times, the display data 140 is insufficient, and so the NE must query the data network, as represented by the path 180 of queries, to further isolate and diagnose the fault. This usually takes the form of scripts, such as, ping, or traceroute, and through the use of sniffers. The network 110 then sends back to the NE query results in path 185.

10  
15

Block 190 illustrates a flow chart of the process engaged by the NE 150 to perform troubleshooting. In step 160, the NE 150 analyses the fault data presented and diagnoses the fault. In step 165, the NE 150 develops and then implements a plan to correct the problem. Thereafter, the NE must verify the problem has been eliminated, in step 170. A NE may submit a report of the incident in step 175.

25 Thus, network diagnosis in the prior art was a manual process conducted by the NE. As Figure 1 illustrates, the NE is responsible for isolating and identifying faults that are causing the problem. This can be time consuming and tedious,

especially for large enterprise networks. As such, the analysis of fault data by the NE in today's larger heterogeneous networks is prone to error due to the large amounts of fault data to be manually processed.

5

Thus, a need exists for lessening the burden on a network engineer in the process of diagnosing faults within a data network. A further need exists for increasing efficiency in the process of diagnosing faults within a data network.

## SUMMARY OF THE INVENTION

The present invention provides a method for self-diagnosing faults within a data network. One embodiment of the present invention provides a method that achieves the above  
5 accomplishment and which also provides for a method and system that lessens the burden on a network engineer when diagnosing faults. Additionally, one embodiment of the present invention provides a method and system that achieves the above accomplishments and which also provides for increased  
10 efficiency in the process of diagnosing faults within a data network.

Specifically, one embodiment of the present invention discloses a method for automating the process of determining  
15 faults in a data network. A network management station (NMS) receives filtered fault data from the various subnetworks of computer systems that comprise the network. Each of the subnetworks have an associated performance manager that monitors, collects and filters fault and network performance  
20 data from the subnetwork. Each of the performance managers send their respective filtered data to the NMS. The NMS further analyses the filtered data to identify the fault and locate a source of the fault within the topology of the network. The NMS can further query the network for additional  
25 fault and network performance data as necessary. The NMS displays the fault, cause of the fault, and location of the fault to a network engineer for correction.

These and other benefits of the present invention will no  
doubt become obvious to those of ordinary skill in the art  
after having read the following detailed description of the  
5 preferred embodiments which are illustrated in the various  
drawing figures.

These and other benefits of the present invention will no  
doubt become obvious to those of ordinary skill in the art  
after having read the following detailed description of the  
5 preferred embodiments which are illustrated in the various  
drawing figures.

## BRIEF DESCRIPTION OF THE DRAWINGS

PRIOR ART Figure 1 is a block diagram illustrating the traditional primary role of the network engineer within a process for diagnosing a fault within a data network.

5

Figure 2 illustrates a block diagram of an exemplary electronic device that is capable of diagnosing a fault in a data network, in accordance with one embodiment of the present invention.

10

Figure 3 is a block diagram illustrating a process for self diagnosing a fault within a data network, in accordance with one embodiment of the present invention.

15

Figure 4 is a diagram illustrating an exemplary data network that is capable of diagnosing faults, in accordance with one embodiment of the present invention.

20

Figure 5 is a block diagram of an exemplary data network that is capable of diagnosing faults, in accordance with one embodiment of the present invention.

25

Figure 6 is a flow diagram illustrating steps in a method for diagnosing faults within a data network, in accordance with one embodiment of the present invention.



Figure 7 is a flow diagram illustrating steps in a method for analyzing fault data in order to identify two exemplary fault types and isolate the source of instances of the faults.

Figure 7 is a flow diagram illustrating steps in a method for analyzing fault data in order to identify two exemplary fault types and isolate the source of instances of the faults.

## DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the preferred  
embodiments of the present invention, a method for diagnosing  
faults within a data network, examples of which are illustrated  
5 in the accompanying drawings. While the invention will be  
described in conjunction with the preferred embodiments, it  
will be understood that they are not intended to limit the  
invention to these embodiments. On the contrary, the invention  
is intended to cover alternatives, modifications and  
10 equivalents, which may be included within the spirit and scope  
of the invention as defined by the appended claims.

Furthermore, in the following detailed description of the  
present invention, numerous specific details are set forth in  
15 order to provide a thorough understanding of the present  
invention. However, it will be recognized by one of ordinary  
skill in the art that the present invention may be practiced  
without these specific details. In other instances, well known  
methods, procedures, components, and circuits have not been  
20 described in detail as not to unnecessarily obscure aspects of  
the present invention.

### NOTATION AND NOMENCLATURE

Some portions of the detailed descriptions which follow  
25 are presented in terms of procedures, steps, logic blocks,  
processing, and other symbolic representations of operations on  
data bits that can be performed on computer memory. These

descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "receiving," or "processing," or "filtering," or "analyzing," or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical

quantities within the computer system memories or registers or other such information storage, transmission or display devices.

5 Referring now to Figure 2, portions of the present invention are comprised of computer-readable and computer-executable instructions which reside, for example, in computer-readable media of a computer system. Figure 2 is a block diagram of exemplary interior components of an exemplary computer system 200, upon which embodiments of the present invention may be implemented.

10 Figure 2 illustrates circuitry of an exemplary computer system 200. Exemplary computer system 200 includes an address/data bus 220 for communicating information, a central processor 201 coupled with the bus 220 for processing information and instructions, a volatile memory 202 (e.g., random access memory (RAM), static RAM dynamic RAM, etc.) coupled with the bus 220 for storing information and instructions for the central processor 201, and a non-volatile memory 203 (e.g., read only memory (ROM), programmable ROM, flash memory, EPROM, EEPROM, etc.) coupled to the bus 220 for storing static information and instructions for the processor 201.

25 Exemplary computer system 200 also includes an optional data storage device 204 (e.g., memory card, hard drive, etc.)

coupled with the bus 220 for storing information and instructions. Data storage device 204 can be removable. Exemplary computer system 200 also contains an electronic display device 205 coupled to the bus 220 for displaying  
5 information to a user. The display device 205 utilized with the computer system 200 may be a liquid crystal device, cathode ray tube (CRT), field emission device (FED, also called flat panel CRT) or other display device suitable for creating photographic or graphical images.

10 Exemplary computer system 200 also contains an alphanumeric input 206 for communicating information and command selections to the central processor 201. In addition, system 200 also includes an optional cursor control or directing  
15 device 207 coupled to the bus 220 for communicating user input information and command selections to the central processor 201.

20 With reference still to Figure 2, an optional signal Input/Output device 208, which is coupled to bus 220 for providing a communication link between computer system 200 and a network environment, is described. As such signal  
25 Input/Output device 208 enables the central processor unit 201 to communicate with or monitor other electronic systems or analog circuit blocks coupled to a data network.

#### DIAGNOSING FAULTS IN A DATANETWORK

Accordingly, the present invention provides a method for diagnosing faults within a data network. Also, one embodiment of the present invention provides a method that lessens the burden on a network engineer for diagnosing faults within a data network. Still another embodiment of the present invention increases the efficiency of diagnosing faults within a data network.

The flow charts 600 and 700 of Figures 6 and 7, in combination with Figures 3, 4, and 5, illustrate a method for self diagnosing network faults within a data network, in the present invention. The Figures 3, 4, and 5 provide a system architecture for implementing the method outlined in flow chart 600 of self diagnosing a data network.

Although embodiments of the present invention are discussed within the context of a data network, other embodiments are well suited for implementation within communication networks that are internet-protocol (IP) based data networks, such as a Transmission Control Protocol/Internet Protocol (TCP/IP) based data network.

Network faults can be classified as hardware and software faults. These hardware and software faults can in turn cause other fault effects in the network such as congestion or even network failure.

Examples of hardware faults are failures of devices due to physical external events, such as, accidents (e.g., kicking a cable loose), mishandling, or improper

5 installation, etc. Other hardware faults are due to the aging of the device and/or weaknesses in the design of the hardware device.

Software faults cause network devices to produce  
10 incorrect outputs. Software faults occur as slow or faulty service by the network due to incorrect information (e.g., erroneous router tables) or erratic behavior of the device due to software bugs, (e.g., incorrect processing of packets). Software faults can also surface as lack of  
15 communication service if software tables have become corrupted (e.g., spanning tree loops that result in no packet forwarding at all).

With reference now to Figure 3 and flow chart 600 of  
20 Figure 6, exemplary steps used by the various embodiments of the present invention are illustrated. Flow chart 600 includes processes of the present invention which, in one embodiment, are carried out by a processor under the control of computer-readable and computer executable instructions.  
25 The computer-readable and computer-executable instructions reside, for example, in data storage features such as computer usable volatile memory 202, computer usable non-

volatile memory 203, and/or data storage device 204 of Figure 2. The computer readable and computer-executable instructions are used to control or operate in conjunction with, for example, central processing unit 201 of Figure 2.

5

In Figure 3, an exemplary communication network 300 is shown that is capable of self diagnosing faults as illustrated by the flow chart 600, in accordance with one embodiment of the present invention. As shown in Figure 3, system 300 is a local area network (LAN), but other embodiments are well suited to other networks having a plurality of remotely located and coupled computer systems. In one embodiment, the communication network 300 is a transmission control protocol/internet protocol (TCP/IP) based data network.

A generalized self diagnosing network (SDN) system 335 controls the process 330 of automating the diagnosis of faults within the data network 300. In the present embodiment, the SDN system 335 receives a plurality of fault data, as illustrated in step 610 of flow chart 600. For example, the SDN system 335 receives alarm data 310 from the data network 300. Alarm data 310 come in the form of alarms or events where preset thresholds within devices of the data network 300 have been exceeded or not been met. In addition, network data 315 consists of performance data that is constantly monitored by a network management tool. In one



present embodiment, the network data 315 consists of the standardized remote monitoring 1&2 (RMON 1&2) data.

The SDN system 335 can also make further queries to the data network 300 for the fault diagnosis process if there is insufficient data. The SDN system 335 sends queries 320 for fault data to devices on network 300. The responses or query results 325 are sent directly back to the SDN system 335 to be combined with the previously collected and received fault data.

Block 330 of Figure 3 illustrates a generalized process for diagnosing faults within a data network 300 by the SDN system 335. The steps illustrated in flow chart 600 are embedded in the diagnosing fault block 340 of Figure 3.

In step 620 of flow chart 600, the present embodiment filters the fault data to eliminate extraneous data. The filtering process significantly reduces the amount of fault data. In the present embodiment, alarm filtering can be accomplished by eliminating multiple occurrences of the same alarm, inhibiting low priority alarms, generalizing alarms to their superclasses (as determined by domain experts), and replacing a specified occurrence of similar alarms with a single "count" alarm. Other embodiments are well suited to the use of other filtering processes.

In one embodiment, the filtering process can be implemented through rules or algorithms. The data appears in the form of alarms, RMON 1&2 data, or any other suitable monitored data.

5

Once alarms are filtered, there is still an enormous amount of data to be analyzed. Coupled with the filtering process of step 620, the present embodiment also correlates the alarms by substituting a new item for a set of alarms that match a predefined pattern. In one embodiment, this process can be approached as a pattern recognition problem.

In one embodiment, alarm correlation requires the speedy analysis of large amounts of fault data. The fault data is quickly classified into recognized patterns, clusters, or categories of fault data. For this task, statistical and probabilistic approaches are appropriate, such as, using neural network analysis, or statistical algorithms. The output takes on the form of problem types and characteristics that are recognizable in troubleshooting the data network 300.

After filtering and correlating the plurality of fault data from the network 300, a core of fault data is produced by the SDN system 335. This core of fault data is further analyzed by the SDN system 335 by the present embodiment, in step 630 of flow chart 600. The analytical process of step

630 detects the fault by identifying the type of fault and isolating the misbehaving device in the network that is the cause of the fault.

5 In one embodiment of the present invention, the analytical process of fault detection in step 630 requires a symbolic approach. In this symbolic approach, the fault data that is filtered and correlated, and other suitable network data, can be analyzed to determine if more data must be  
10 obtained (e.g., obtained through the use of scripts, such as, ping or traceroute).

In another embodiment, the fault diagnosis process as outlined in flow chart 600 is an iterative process. In other  
15 words, when additional and different data is needed from the data network 300 to assist in diagnosing the fault, the present embodiment sends requests for more fault or network performance data, receives the answers to those request, and continues to troubleshoot the problem using the newly  
20 acquired information.

Referring now to Figure 7, flow chart 700 illustrates steps in the analytical process that is outlined in step 630 of Figure 6, in accordance with one embodiment of the present  
25 invention. From step 630, the present embodiment implements the analytical process by determining if the core of fault data is due to two example faults: a broken link or

congestion within the data network 300, in condition block 710.

If the core of fault data is due to a broken link within the network 300, then the present embodiment via SDN system 335 performs a ping walk in step 720. Those well versed in the art understand that the function of the ping walk is to traverse the network topology (or map) generated by a networking monitoring tool by pinging all the Internet Protocol (IP) addressed devices in the network 300 and identifying those that are not reachable from anywhere.

In step 730, after the ping walk is performed, the results of the ping walk are compared to the network topology to determine the location of a broken link. In one embodiment, the network topology is produced by a network monitoring tool that produces a tree representation of the data network 300 which uses arcs and nodes to represent the network topology. Those well versed in the art understand that the arcs represent pings and the nodes represent devices on the network.

If however, the SDN system 335 determines that the core of fault data is due to congestion, then flow chart 700 proceeds to step 740. Typically, congestion occurs when a device or a link in the network is experiencing delay or packet loss due to congestion conditions such as, the load on

the network or a bottleneck. Congestion is detected by examining traffic data that is monitored. Congestion collapse of a network occurs when the total productive work done by a network decreases significantly with an increase in the load on the network.

The traffic data is comprised of delay, packet loss, and queue size, in one embodiment. A delay in a network is an increase in time a packet has to wait at a network device before being forwarded. Packet loss is the percentage of packets which are dropped due to queue overflows. Queue size is the length of the packets queued at a network device. Queue size gives the earliest indication of congestion, followed by delay, and then by packet loss. If congestion is detected early, often only small adjustments are needed to cure the congestion.

For example, delay is commonly measured by looking at the round trip time of packets in the network. Those well versed in the art understand that pinging is a useful tool in probing links for congestion, in one embodiment. On end nodes of a network which are communicating via TCP, the round trip time maintained as a baseline target is an excellent direct measure of network delay. By pinging various internet protocol (IP) devices addressed in the network, possible problems due to delay may be alerted to the NE and averted.

In step 740, the present embodiment analyses traffic data to determine the identity of the fault, due to congestion, that is associated with the core of fault data. Traffic data is RMON 1&2 data, or any other monitored network data, such as, delay. In step 750, the present embodiment analyses the traffic data to isolate the source of the core of fault data.

In step 740 and 750 of flow chart 700, in one embodiment the process of deductive reasoning is used to determine the type of fault, the cause of the fault, and the location of the fault. In one embodiment, the deductive process used to identify and isolate the fault is implemented using fuzzy pattern matching technologies such as case-based reasoning or deductive/inductive logic techniques such as expert systems..

Referring now back to Figure 3, in block 345 of the generalized process for fault diagnosis of block 330, the SDN system 335 displays the results (e.g., identifying the type of fault, the cause, and the source of the fault) to the network engineer with an explanation of its reasoning, in accordance with one embodiment of the present invention. In another embodiment, a short description of possible remedial or corrective actions is developed and displayed to the NE as shown by block 347.

The SDN system 335 as illustrated in block 330 of Figure 3 illustrates the easing of the burden on the NE 150 by automating the fault diagnosis process. This is done by automating the analysis burden. However, in another embodiment, the NE continues to retain as much control over the automation process as is needed.

As shown in Figure 3, the NE receives information at a much later point than in the traditional fault diagnosis process as illustrated in Prior Art Figure 1. The burden of analyzing the large amounts of fault data is now provided by the SDN system 335. The NE can step in later in the fault diagnosis process and utilize the information provided by the SDN system 335 regarding fault identification and source isolation to take an appropriate corrective action, in block 350. After implementing a remedy, the NE can further verify fault elimination in block 355. And in block 357, the NE can provide documentation of the fault and the remedies implemented.

Figures 4 and 5 are diagrams illustrating exemplary architectures of self diagnosing networks (SDN) systems. Figure 4 gives an exemplary architecture of a SDN system 400. Figure 5 gives a block diagram of an exemplary SDN system 500 that is representative of the architecture of SDN system 400.

Referring now to Figure 4, the SDN system 400 is comprised of a multi-layered network. The lower layer is comprised of a plurality of subnetworks (subnets), e.g., subnet-1 410 on up to subnet-n 420. The plurality of scaleable subnetworks create the data network (e.g., the network LAN 300 of Figure 3).

Located within each of the subnetworks (e.g., subnet-1 410 on up to subnet-n 420), are edge devices and core devices. The edge devices are comprised of electronic systems, such as, computer systems, printers, facsimile machines, etc. In subnet-1 410, the edge devices are 410a, 410b, on up to 410n. In subnet-n 420, the edge devices are 420a, 420b, on up to 420n. Each of the edge devices are capable of collecting RMON 1&2 data for network performance and fault monitoring that is used for fault analysis, in one embodiment.

The core devices within the subnetworks (410 on up to 420) are comprised of network support devices, such as, routers, switches, hubs, etc. Most of the core devices are also capable of collecting RMON 1&2 data for fault monitoring and fault analysis, in one embodiment. In subnet-1 410, the core device 415 is coupled to each of the edge devices, 410a, 410b, on up to 410n. In subnet-2 420, the core device 425 is coupled to each of the edge devices, 420a, 420b, on up to 420n.



The RMON 1&2 data collected in each of edge devices of subnet-1 410 and subnet-2 420 are passed on up to their respective core device and then sent to a middle layer edge monitor. For example, in subnet-1 410, the RMON 1&2 data is sent to the edge monitor 430a. In subnet-2-420, the RMON 1&2 data is sent to the edge monitor 430b.

Each of the edge monitors at the middle layer provide the filtering and correlating of the RMON 1&2 fault data produced by each of the edge and core devices located in their respective subnets, as illustrated in step 620 of the fault diagnosis process of flow chart 600.

After the fault data is reduced to a core of fault data by filtration and correlation, the core of fault data is transferred to the top layer. The top layer is comprised of a network management station 440 that is coupled to each of the subnetworks (e.g., subnetwork-1 410 on up to subnetwork-n 420) via their respective edge monitors (e.g., 430a and 430b). The network management station 440 provides the analysis portion of the SDN system 400, as illustrated in step 630 of flow chart 600. In general, the station 440 troubleshoots any problems experienced by the network 400, further analyzes the core of fault data, and reports potential problems and solutions to the network engineer managing the network 400.

With reference to Figure 5, the SDN system 500 is similar to the architecture of Figure 4, in accordance with one embodiment of the present invention. The SDN system 500 is comprised of three layers, a lower layer 510, a middle layer 520, and a top layer 530. The SDN system 500 is capable of diagnosing faults within the communication system or network 550.

In comparison to Figure 4, the lower layer 510 of Figure 5 is analogous to the core and edge devices that comprise each of the subnetworks (e.g., subnet-1 410 and subnet-n 420). In Figure 5, the lower layer is comprised of performance managers (e.g., 512, 514, on up to 517). Each of these performance managers devices supply management information base (MIB) information to the SDN system 500, such as RMON 1&2 data, in one embodiment. Other embodiments are well suited to other MIB tables to be incorporated in to the diagnostic system.

The middle layer 520 of Figure 5 consists of SDN network performance managers (SDNnpm), such as, SDNnpm 522, SDNnpm 524, and SDNnpm 527. In comparison to Figure 4, the middle layer 520 of Figure 5 is analogous to each of the edge monitors (430a and 430b) that are associated with respective subnetworks. The role of the SDNnpm (e.g., 522, 524, and 527) within the SDN system 500, is to poll the various

coupled performance managers devices (e.g., 512, 514, and 517 respectively) for MIB data, filter and correlate those data (e.g., step 620 of flow chart 600), and transfer the remaining data that has been filtered and correlated to the top layer 530.

The top layer 530 of Figure 5 consists of a SDN network management system (SDNnms) 535. In comparison to Figure 4, the SDNnms 535 is analogous to the top network management station 440. The SDNnms 535 acts as the central control of the SDN system 500 and performs the fault diagnosis of the data network 550, as illustrated in step 630 of flow chart 600.

In addition, the SDNnms 535 interacts with a network topology system 540 for topology information regarding the network 550. The topology system 540 discovers the topology of the network 550 and supplies this topology to the SDNnms 535 in graphical form. Thereafter, the SDNnms 535 is able to perform fault diagnosis of the network 550 utilizing the network topology to isolate the source of faults due to broken links and congestion.

While the methods of embodiments illustrated in flow charts 600 and 700 show specific sequences and quantity of steps, the present invention is suitable to alternative embodiments. For example, not all the steps provided for in

the method are required for the present invention.

Furthermore, additional steps can be added to the steps presented in the present embodiment. Likewise, the sequences of steps can be modified depending upon the application.

5

A method for diagnosing faults within a communication network, is thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.

10